



SELinux

Arabic Guide



م / صبري صــالح



SELinux Arabic Guide

إهداء:

في هذا الكتاب إهداءين/
إهداء خاص لإخواني: د/علي الشمري(B!n@ry), د/ بشار حامد(باحث عن المعرفة), أحمد حسن(Dr.Hacker) , محمود(No4hard) , محمد العتيبي Bad3r , إسلام فكري , إسلام اليماني(The Ghost),رامي علام , و لكل من يحبونني و أحبهم في الله.
إهداء عام: لجميع المسلمين و من سينتفع بهذا الكتاب في ما يرضي الله عز و جل.

رخصة الكتاب:

الكاتب: صبري صالح (KING SABRI)
الموقع: مجتمع الحماية العربي
للمراسلة: Sabri@Security4Arabs.net

الـمقدمة:

أبدء كتابي باسم الله الرحمن الرحيم, هذا الكتاب الأول على المستوى العربي في ما يتعلق بنظام الـ SELinux. قررت كتابته لإزالة التخوف من الدخول فيه و تعلم هذا النظام القوي و الذي ينير لنا طريق من طرق الحماية الحديثة و أيضا لتبيين قوة نظام GNU Linux عن سائر الأنظمة و للتقليل من المشاكل الأمنية التي تواجه منظماتنا.

ملاحظة: افترضت في هذا الكتاب أن القارئ مُلم بالتعامل مع النظام و تصاريح الملفات و الخدمات بشكل قوي. كما أنصح القارئ بعدم الاستعجال في القراءة فأي نقطة غامضة سيتم توضحيها في النقاط القادمة لأنه من المستحيل توضيح كل شئ في آن واحد.

كما أحب أن أنوه أن هذا المجهود ليس عبارة عن ترجمة كتاب.

الـمصادر:

تم فهم واقتباس معلومات هذا الكتاب من مصادر متعددة

RH033 , RH133 , RH253, RHS429 , fedoraproject.org , engardelinux.org , google.com

الفهرس

الباب الأول: مقدمة عن SELinux

نبذة عن SELinux

نظام التصاريح التقليدي DAC

نظام الـ SELinux – MAC

ما يستطيع نظام SELinux عمله

ما لا يستطيع نظام SELinux عمله

بُنية SELinux

شرح User Identity و Role

شرح الـ Domain / Type

شرح الـ Sensitivities و Categories

طريقة كتابة جمل الحماية Security Context

ما هي السياسة/المنهاجية في النظام (SELinux Policy)

السياسة الموجهة (Targeted Policy)

السياسات المنطقية (Policy Boolean)

عرض الـ Security Context

الأرشفة و ضغط الملفات

تدريب عملي شامل

الباب الثاني: استخدام الـ SELinux

أوضاع الـ SELinux

التحكم بالـ SELinux

سياق الملفات (File Context)

إعادة سياق الملفات (Relabel Files)

إعادة سياق نظام الملفات (Relabel a filesystem)

عمل mount مع الـ SELinux

الباب الثالث: السياسة الموجهة(خاص بـ ريدهات)

الخدمات المحمية بالـ Targeted Policy

التحكم في حماية الخدمات

خدمة الأباتشي – Apache

محتويات الأباتشي

إعدادات منطقية خاصة للأباتشي

خدمة أسماء النطاقات DNS

جمل الحماية و الجمل المنطقية للـ DNS

خدمات أخرى

جمل الحماية لخدمات أخرى

الباب الرابع: التتبع و حل المشاكل Troubleshooting

تحديد المشكلة

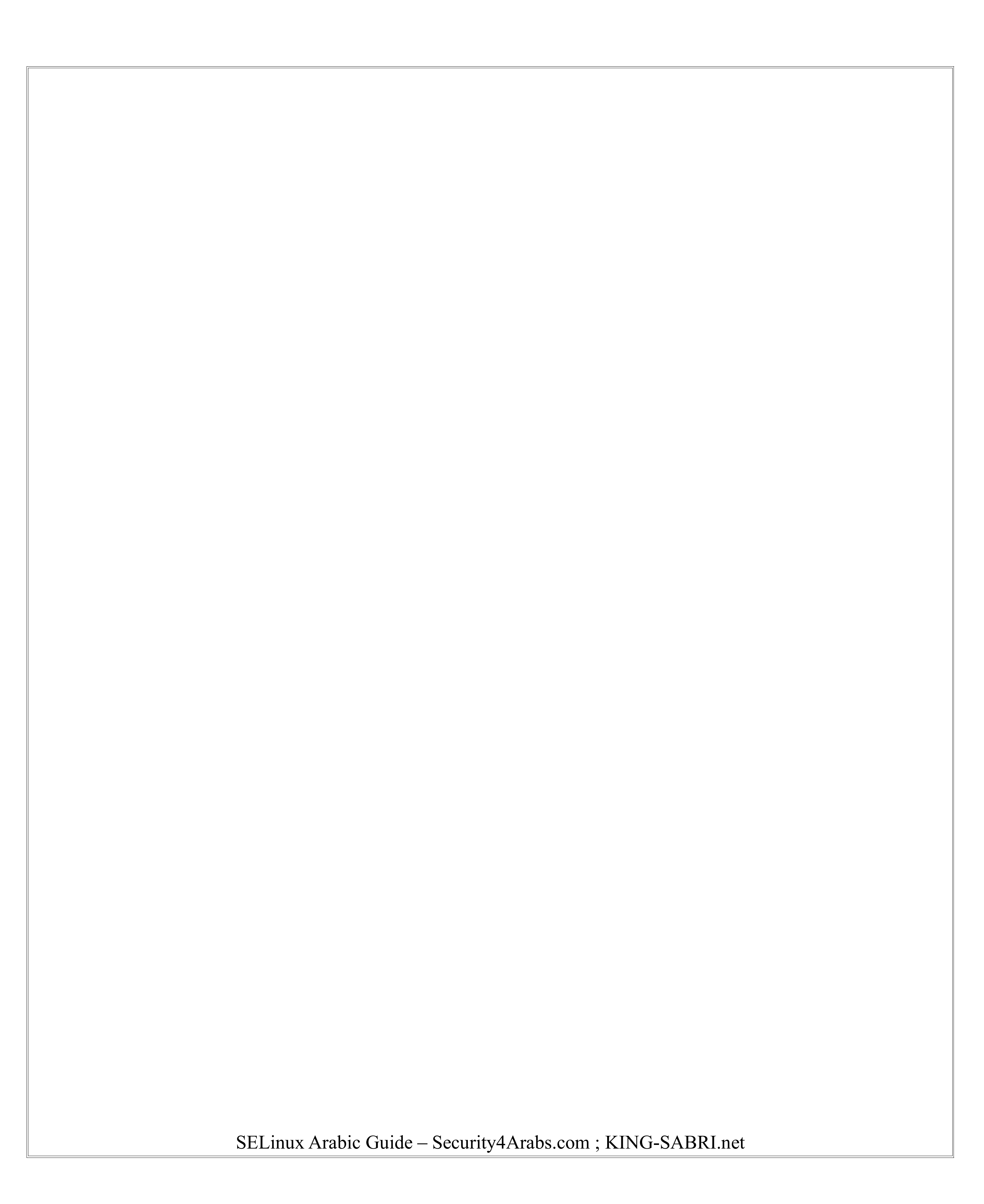
التعامل مع السجلات SELinux Auditing

التعامل مع AVC

حل المشاكل Troubleshooting

الباب الخامس: أدوات سياسات الحماية (برامج إضافية)

الخاتمة





الباب الأول: مقدمة عن SELinux

نـبـذة عن SELinux

مع زيادة الحاجة إلى البرامج و الخدمات في الأنظمة ظهرت زيادة ثغرات الأنظمة و التي تتسبب في تصعيد الصلاحيات (escalating privileges) من مستخدم عادي إلى المستخدم الجذر أو مستخدم ذو صلاحيات أعلى. تكون دائما بسبب ثغرة في خدمة أو برنامج أو طريقة تعامل النظام مع شيء معين و تُعرف بـ(Remote root/Local root) و يقوم المخترق باستغلال الثغرة عن طريق BuffierOverFlow أو حقن أوامر و غيرها من الطرق و جميعنا يعرف أنه كانت هناك ثغرات خطيرة للـ Netcat, NetworkManger, BIND, curl, rsync , apache و غيرها الكثير.

و بناء على ما سبق ذكره,قامت حكومة الولايات المتحدة بأسناد مهمة تطويرية إلى وكالة الأمن القومي(National Security Agency – NSA) لنظام يحتوي قواعد للتعامل مع الملفات و العمليات و طريقة الاتصال في ما بينهم و هذا بناء على أبحاث أثبتت أن أخطر الثغرات تكمن في أن المستخدم العادي يستطيع تخطي حماية النظام الداخلية و كأخطر مثال ما ذكرناه آنفا , أيضا إعطاء التصريح 777 لملفات تنفيذية مهمة و هكذا.

في البداية قامة وكالة الأمن القومية بتطوير نظام إسمه "Mach" على نظام التشغيل "Flask" و كان قاعدته مبنية على عزل البرامج/العمليات و الملفات التنفيذية و المستخدمين عن الملفات و عدم السماح للقسم الأول بالتعامل مع القسم الثاني بشكل ارتجالي و تم تسمية فكرة العزل بـ Type Enforcement -TE.

وجدت الوكالة أن هذه الطريقة فعالة و قوية فقررت دمج هذا النظام في داخل نظام التشغيل اللذي سيعمل فيه و قد اختاروا نظام تشغيل مفتوح المصدر فقاموا بإضافة الرقع إلى الـ Linux Kernel (بعد اذن لينوس تروفالدز) و إسناد حزم (Linux Security Modules - LSM) و حينها تم تغير اسم التقنية التي يعمل بها إلى Mandatory Access Control -MAC و تم تسمية المشروع بـ Security Enhancement Linux -SELinux و هو الآن متوفر في أغلب/جميع التوزيعات بشكل افتراضي و يعمل على نظام ملفات ext2,ext3,ext4.

هناك مقولة مشهورة في عالم اللينوكس تقول:

Every Thing is a **File** – كل شيء يُعتبر **ملف**

حيث الوصول للملف يخضع إلى تصاريح تقليدية يتحكم بها و المالك سواء كان User أو Group.

أما بالنسبة لـ SELinux فالمقولة تصبح:

Every Thing is an **Object** – كل شيء يُعتبر **موضوع**

حيث الوصول للموضوع يخضع إلى مجموعة عناصر تخضع تلك العناصر إلى قوانين يتم تطويرها باستمرار حيث هذه العناصر تسمى (Security Context) و مجموعة القواعد الموجودة تسمى Policy و هي متجددة أيضا باستمرار.

بعد ظهور SELinux تم تصنيف حماية الملفات إلى تصنيفين رئيسين:

1. Discretionary Access Control – DAC

و هي تقنية تصاريح الملفات التقليدية للينوكس. في هذا النظام جميع العمليات(processes) تعمل تحت User و Group مع نظام DAC تستطيع تلك العمليات الوصول إلى جميع الملفات و المجلدات التي يستطيع مستخدمها أو مجموعتها الوصول لها و عند حدوث أي خطأ/ثغرة في هذه العمليات يمكن أن تدمر جميع البيانات التي تصل لها هذه العملية و هناك عمليات لها أحقية الوصول إلى أماكن خطيرة في النظام.

هناك فقط نوعين من أنواع المستخدمين:

1) Administrators/Privilege-users

2) Non-Administrator/Non-privilege-users

إن الخدمات و البرامج (العمليات بشكل عام) قد تتصاعد من مستخدم عادي إلى مستخدم مدير عند حاجتها لذلك و قد تصل إلى مستوى صلاحيات الجذر نفسه لتقوم بعملها على أكمل وجه.

أيضا من عيوب هذا النظام احتمالية كبيرة في الخطأ عند إسناد تصاريح الملفات عند استخدام الأمر المفا chmod أو استخدام setuid,setgid بشكل خاطئ إذن فالملفات فقط هي التي تأخذ تصاريح.

2. Mandatory Access Control – MAC أو Non-DAC

في هذا النظام يتم إسناد تصاريح لـ **<u>كيف ستتعامل كل العمليات(Process)-كل واحدة على حدى- مع أجزاء النظام الأخرى مثل الملفات و الأجزاء المادية و المنافذ و العمليات الأخرى</u>.**

هذا المبدأ يتم تنفيذه عبر سياسات الحماية (Policeies) حيث توضع قوانين لكل الخدمات/العمليات و قوانين لكل الملفات تحت قواعد تحدد سماحية الوصول إليها.

مثال على ذلك

المستخدمين الذين عرضوا بياناتهم للوصول في مجلدهم الرئيس باستخدام الأمر chmod فإنه سيتم حماية تلك البيانات من العمليات التي قد تصل إليها لأنهم يخضعون لـ policy خاصة بملفات/مجلدات المستخدمين. فقط ستستطيع العمليات الوصول لهذه الملفات عند تغير الـ policy الموضوعة لمجلد المستخدم.

هام!!

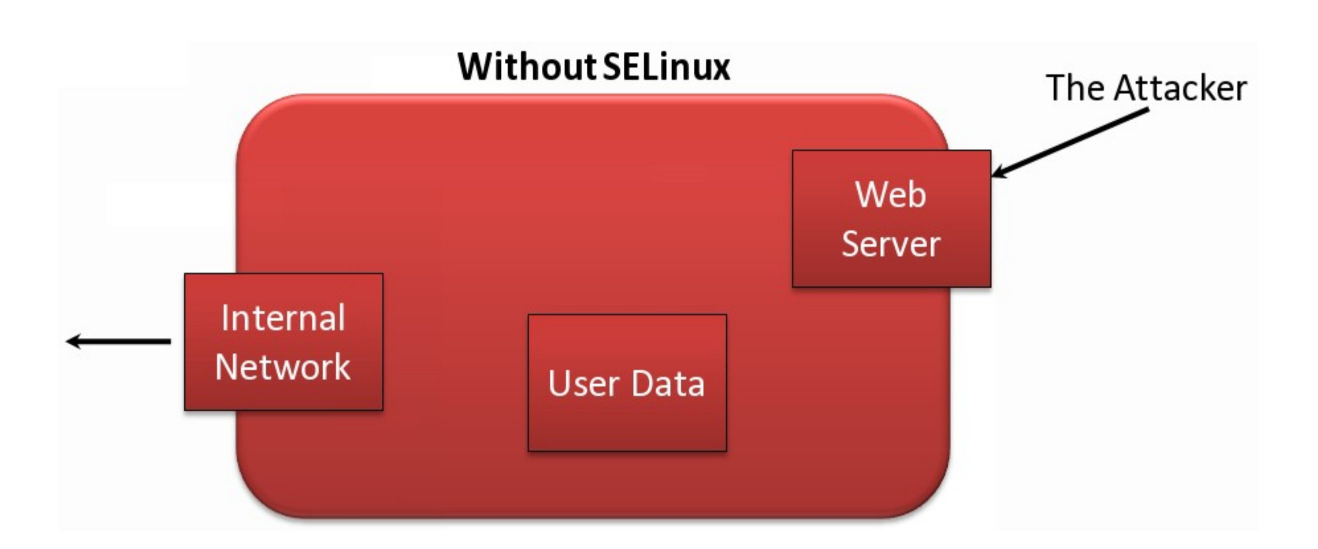
- هناك تسميات خاصة بالـ SELinux بالنسبة للعمليات و الملفات, و هي كالتالي:

-- العمليات(process)/الخدمات(service)/المستخدمين(user) تسمى: **Subject**

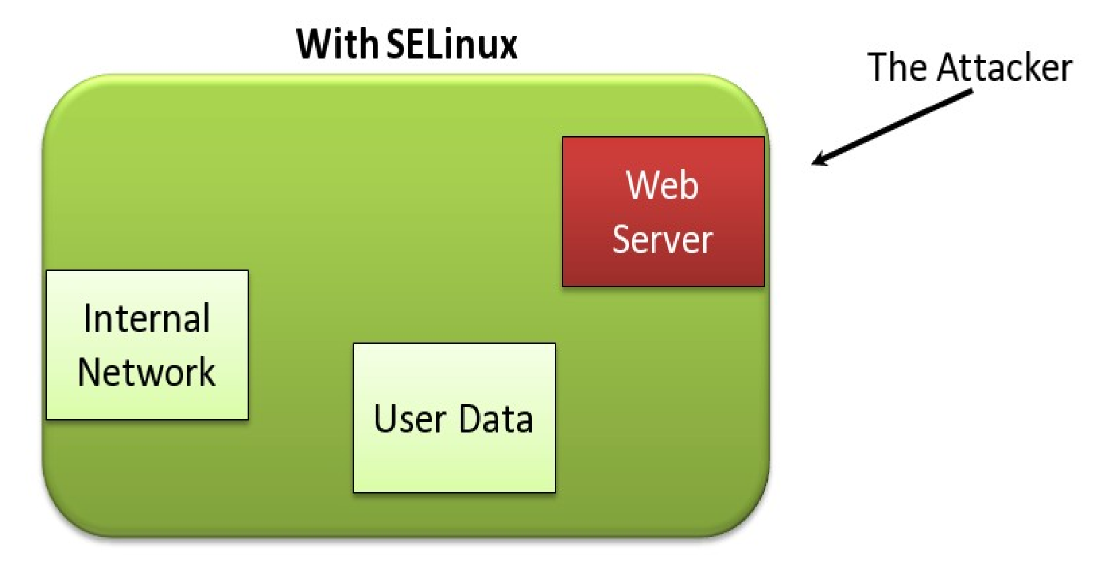
-- الملفات(files)/الأجهزة(device)/المقابس(sokect)/المنافذ(port) تسمى: **Object**

<u>سيتم استخدام هذه المصطلحات من الأن و حتى نهاية الكتاب فأرجوا ألا تنساها هي و كل ما المصطلحات التي سيتم ذكرها و تعريفها.</u>

رسم توضيحي يبين كيف يفصل الـ SELinux الـ Subject عن الـ Object فائدته



في نظام الـ DAC لو استطاع المخترق أن يستغل ثغرة للوصول إلى النظام و استغل ثغرة في النظام لتصعيد صلاحياته(Local root exploit) فإنه من الطبيعي أن سيضر ببيانات المستخدمين و يحاول الوصول إلى الشبكة الداخلية و بدء هجوم آخر.



مع نظام الـ MAC, لو استطاع المخترق أن يستغل ثغرة للوصول إلى النظام فإنه لن يستطيع الوصول إلا لما تستطيع خدمة الـ httpd الوصول إليه فقط و ما يندرج تحت سياسة الوصول لها فقط و سيضل محبوسا في هذا النطاق.

ما يستطيع أن يفعله SELinux

مهمة SELinux الكبرى هي هي فرض تقنية الـ MAC policy و التي تقيّد البرامج و العمليات و الخدمات في أقل قدر من الصلاحيات المطلوبة لإتمام عملها بنحاج.

يوفر النظام الحماية من توسع استغلال الثغرات عن طريقة النقاطة العامة التالية:

- منع تصعيد الصلاحيات
- منع القراءة و الكتابة والتعديل الغير مصرح له على البيانات من قبل البرامج/العمليات/الخدمات
- تسجيل أحداث الدخول و التجاوزات
- سهل التعديل و إضافة القوانين
- يوفر الـ Type Enforcement
- يوفر الـ role- based access control

ما لا يستطيع أن يفعله SELinux

الـ SELinux حاله كحال أي نظام حماية لا يستطيع فعل كل شئ في كل مكان فهوء جزء/طريقة لزيادة الأحماية و ليس للحماية المطلقة.

- لا يستطيع حماية كل أجزاء النظام
- لا يستطيع الإحاطة بكل العملياتو الخدمات (لهذا يتم تحديث الـ policy باستمرار)
- لا يغني عن الجدار الناري
- لا يغني عن حماية الخدمات الحماية المطلقة
- لا يغني عن أهمية تحديث الحزم و النظام و الخللات الأمنية
- وجوده لا يعني الإهمال في كتابة الكود و عندم الانتباه لنقاط الضعف في البرامج التي نكتبها

ملاحظة: يتم تطبيق نظام الـ MAC بعد تطبيق الـ DAC أي أنه إذا قام نظام DAC بمنع الوصول فإن النظام لا يقوم بالنظر إلى الـ DAC لأن المطلوب-وهو الحظر/المنع- قد تم أم إذا كان التصريح يسمح بالوصول فإن النظام يقوم بالنظر إلى سياسة الـ MAC و يطبقها و هذا لا يقلل من قوة الـ MAC بل يزيده قوة و تميز فبدلا من أن يكون هناك حاجز واحد أصبح هناك حاجزين.

بُـنية SELinux

في بيئة SELinux تعمل Process أو (Object) في Domain حيث يسجن/يقيّد الـ processes المعروفة لديه و يُعرف هذا القيد أو السجن بالـ(Sandbox) و الذي يتم فيه ضبط و تعريف طريقة وصول العمليات إلى الملفات أو (Subjects). تعمل تلك العمليات تحت سلطة المستخدم الخاص بها و تسمى هذه السلطة(Role). تحدد الـ Role أي الـ Process يجب أن تكون/تندرج تحت أي Domain.

بنية SELinux تعمل بناء على شيئين رئيسين:

1. جمل الحماية – Security Context

و تتكون من:

- User Identity
- Role
- Domain / Type
- Sensitivity
- Category

2. سياسة الحماية – Security Policy

كل Object و كل Subject يمتلك Security Context. و كل Security Context يجب أن تحتوي على ثلاثة معاملات أساسية

1) User_Identity ,

2) Role

3) Domain/Type

و تظهر على شكل أعمدة يفصل كل عمود منهم علامة " : " هكذا

```
user_identity:role:type:sensitivity:category
```

مثال:

```
user_u:object_r:httpd_sys_content_t
```

بقية معاملات جمل الحماية ليست شرطا في وضع الـ Targeted Policy سيتم توضيح ذلك لاحقا.

أما عن الـ Policy ,, فهي عبارة عن مجموعة من القواعدة Rules توجه محرك الـ SELinux في تطبيقه للنظام ككل و سنشرح ذلك لاحقا إن شاء الله.

شرح User Identity و Role

إن الـ User Identity في الـ SELinux يكون مقيض بـ Role واحدة أو أكثر, حيث الـ Role تكون بمثابة صلاحيات هذا المعرّف و الذي لا يستطيع تجاوزها. كما نستطيع نقله من Role إلى أخرى بكل سهولة عن طريق الأمر `newrole` و هذا الأمر يشبه في عمله الأمر `su` حيث ينقله من صلاحيات مستخدمه إلى صلاحيات مستخدم آخر.

إن من أهم ما تعمله الـ Role هو تحديد أي مستخدم ينتمي إلى أي Process حيث في الـ Role تكمن صلاحيات الـ Processes/users

يتم تخزين الـ Role و UID في قاعدة بيانات و يكون شكلها عند العرض كالتالي:

ينتهي الـ UID بـ **_u**

مثال:

system_u

تنتهي الـ Role بـ **_r**

مثال:

object_r

لم ننتهي من Role فهناك ما يجب معرفته في الفصول القادمة بإذن الله.

شرح الـ Domain / Type

ذكرنا مرارا أن الـ Subjects يتم وضعها في Sandbox يُسمى الـ Domain و الأن ظهر أمامنا مصطلح جديد و هو Type و يعتبر مقرونا به.

فما هو الفرق بين الـ Domain و ما هو الـ Type و ما هو الـ Sandbox باختصار؟

سأضع تعريفات بالترتيب إفهمها و احفظها جيدا لأن هذه المفاهيم ستسمر معنا إلى النهاية

الـ **Sandbot** : هو السجن الذي يوضع فيه الشئ بغرض عزله أو تقيّده أو حمايته.

الـ **Domain** : هو الـ Sandbox المخصص للـ Subjects أيا كان نوعها

الـ **Type** : هو الـ Sandbox المخصص للـ Objects أيا كان نوعها

إن الـ Policy في الـ SELinux تحتوي على عدة users و roles لكنها تحتوي على مئات أو ألاف من الـ type.

هذه الـ Policy تحدد أي من الـ Domain له أحقية الوصول إلى أي Type .

عند كتابة الـ domain أو الـ type فإنه يظهر بهذا الشكل: **_t**

مثال:

httpd_t

شرح الـ Sensitivities و Categories

هذه الجزئية تم ذكرها في نقاط في موضوع بنية SELinux و لم نتطرق إليها, لماذا؟

لأن الـ Sensitivity و Category تفيدنا **غالبا** عندما تعمل الـ Policy في الـ SELinux في وضع **strict** و **mls** و سنوضحهم بشكل خفيف عند التطرق لأوضاع الـ SELinux .

هذا الكتاب يشرح عمل الـ SELinux مع السياسة Targeted Policy فقط

طريقة كتابة/سياقة جمل الحماية Security Context

قلنا سابقا(بنيئة SELinux) أن كل Object و كل Subject يمتلك Security Context .

جمل الحماية للـ Object و Subject تعتبر بمثابلة **"لاصقة مطبوعة على جبين كل واحد منهما" تسمى"Label"** يتم تخزينها في ما يسمى Extented Attribute xattrs – أو المعامل الإضافي. يقوم الـ xattrs بإضافة بياناته إلى خصائص الملفات كقيمة يتم قرائتها عند التعامل مع هذه الملفات و نظام ملفات لينوكس يدعم ذلك بشكل كامل الأن.

للاستزادة عن الـ xattrs إقرأ وثائق الأمر attr .. نفذ الأمر التالي:

```
man 5 attr
```

يتم كتابة الجمل كما سبق أن ذكرنا بناء على uid, role, type

سنوضح شكل الكتابة بناء على نوعية الـ uid و نوعية الـ role و نوعية الـ domain/type.هكذا:

-> User Lable:

- Non-privileged User: user_u
- Privileged User (root): root

-> Role-Based Access Control - RBAC

- Non-privilieged User: system_r
- Privileged User (root): system_r

-> Type(Objects(files))/Domain(Subjects(processes/programs/users))

- Privileged/Non-privileged Users: unconfined_t
- Processes ex. {httpd: httpd_t ; dhcpd: dhcpd_t}

يتم تغير الـ Label بالأوامر عن طريق الأوامر التالي: (chcon , restorecon, fixfiles)

أمثلة على أشكال الـ Labels

```
root:object_r:user_home_t
system_u:object_r:httpd_exec_t
user_u:object_r:user_home_dir_t
user_u:object_r:httpd_sys_content_t
system_u:object_r:tmp_t
```

قد يخطر ببالك سؤال: من الذي يحدد هذا الشكل في تسمية الجمل(Label)

الجواب, تسمية طريقة تسمية الجملة موجودة في ملف في المكان التالي:

`/etc/selinux/targeted/contexts/files/file_contexts`

ما هي السياسة/المنهاجية في النظام (SELinux Policy)

الـ Policy : هي مجموعة من القواعدة Rules توجه محرك الـ SELinux في تطبيقه لمنهاجية عمل لنظام الـ SELinux ككل.

حيث تقوم الـ Policy بإخبار الكيرنيل بأي من المكونات تنتمي إلى Subject أو الـ Object .

ملاحظة: السياسة -بشكل عام- تُطبق على جميع المستخدمين و حتى المستخدم الجذر "root".

مثال:

عند إنشاء ملف أو مجلد في مجلد المستخدم فإن الـ Policy تحدد الـ Label لهذا النوع حيث ستعرفه بأنه

- مجلد/ملف = Object - سيكون الـ sandbox له هو type

- بناء على المنهاجية المحددة - و الموجودة مسبقا- لهذا الـ Object ستضع الـ type المناسب في الـ Label الخاص بهذ الـ Object و سيكيون user_home_t

- الأن ستأتي الـ Role لـ UID و تقوم بعملها وهو تحديد أي من المستخدمين/العمليات/الخدمات تستطيع الوصول لهذا الـ Object

مثال:

عند تنصيب خادم الويب apache فإن الـ Policy ستحدد الـ Label لهذا النوع حيث ستعرفه بأنه

- خدمة/عملية/مستخدم = Subject – سيكون الـ sandbox له هو domain

- بناء على المنهاجية المحددة - و الموجودة مسبقا- لهذا الـ Subject ستضع الـ domain المناسب في الـ Label الخاص بهذ الـ Subject و سيكيون httpd_t و سياخذ الملف الثنائي domain فرعي من السابق إسمه httpd_exec_t

- تأتي الأن الـ Role لـ UID و تقوم بتحديد الـ type للملفات/المجلدات للـ httpd و التي يستطيع أن يصلها و يتحكم بها و ستكون من نوع httpd_sys_content_t.

- إن اختلف الـ type الخاص بالمفات/ المجلدات عن httpd_sys-content_t أو httpd_user_content_t أو أي type آخر ينتمي للـ Policy المحددة لـ httpd فإن المستخدم/العملية/الخدمة httpd لن تستطيع الوصول لهذا الملف/المجلد

السياسة الموجّهة (Targeted Policy)

إن السياسية الموجهة الإفتراضية في Redhat و منتجاتها و ما بُني عليها هي Targeted. هنا باختصار لن أقوم بإعادة تعريف الـ Policy لأننا عرّفناه وعرفناه في النقظة السابقة. ما سأقوم بتوضيحة هو مفهوم كلمة موجّهة(Targeted)

كلمة موجهة جائت لأن السياسة تطبق على خدمات **محددة** معروفة مسبقا فكل الـ Subjects و الـ Objects تعمل Domain عام و هو

unconfined_t باستثناء الـ Subjects و Objects المعرّفة/المحددة/الموجهة في الـ Policy حيث هي فقط المحمية.

إن كل ما يندرج تحت الـ Domain المسمى unconfined_t يعتبر غير محمي بنظام MAC ولكن يضل نظام DAC يُطبق عليه.

أين يوجد ملف الـ الـ Policy في النظام ؟ و كيف أعرف إصدار هذه الـ Policy؟

- ملف الـ Policy هو ملف تنفيذي يوجد في المسارالتالي

```
/etc/selinux/targeted/policy/
```

- إصدارالسياسة يظهر في اسم ملف يسمى Pilocy حيث يظهر بهذا الشكل

Policy.PolicyVersion

مثال:

```
/etc/selinux/targeted/policy/policy.21
```

أو اعرض محتويات الملف

```
cat /selinux/policyvers
```

عرفنا أنه الإصدار **21** .

كيف أعرف الخدمات المعّرفة لدى الـ Targeted Policy الموجودة على جهازي؟

بشكل عام جميع الخدمات الافتراضية و/أو المشهورة في النظام تعتبر معرّفة. و يتم تزويد و تحديث الـ Policy باستمرار لجعلها تضم أكبر قدر من الخدمات.

سأسرد الخدمات المشهورة هنا

dhcpd_t

httpd_t

initrc_t

ldconfig_t

mysqld_t

named_t

ndc_t

nscd_t

ntpd_t

pegasus_t

portmap_t

postgresql_t

snmpd_t

squid_t

syslogd_t

winbind_t

ypbind_t

بالطبع مع يخرج عنهم سيعرف بـ unconfined_t كما ذكرنا

السياسات المنطقية (Policy Boolean)

لتسهيل التعديل على الـ policy الموجودة لديك, تم عمل طريقة سريعة تعمل بشكل منطقي (0 أو 1) افتراضيا تكون "0" يتم تخزين هذه القيم في ملفات جاهزة في المسار التالي:

```
/selinux/booleans/
```

حيث يوجد فيه جميع المفاتيح السريعة للتعديل على الـ policy الحالية دون الحاجة إلى عمل Policy جديدة لك.

كل ملف من الملفات يحتوي على قيمتي ثنائيتين فقط تأتي بهذا الشكل " 0 0 " القيمة اليسرى لحالة الخدمة في SELinux و القيمة اليمنى وضع الـ SELinux لهذه الخدمه (pending) في ملف selinux/commit_pending_bools/

لتسهيل الأمر عليك, يوجد أمر (setsebool -P) يقوم بهذه المهمة لك شريطة أن تعرف **إسم القيمة** التي تريد تغيرها

مثال: نريد أن نسمح للأباتشي أن يقرأ ملفات من مجلد المستخدم.

```
setsebool -P httpd_enable_homedirs on
```

و تستطيع أن تستخدم الأرقام الثنائية (0 و 1) حيث 0=لا و 1=نعم ,, (قد تجد سياسات الأصل فيها النفي فانتبه نفي النفي إثبات)

```
setsebool -P httpd_enable_homedirs 1
```

تخزن تلك التعديلات في الملف التالي:

```
/etc/selinux/targeted/modules/active/booleans.local
```

أيضا تستطيع عمل هذا عن طريق الواجهة الرسومية من الأمر

```
system-config-selinux
```

عرض الـ Security Context

بالطبع يجب أن نعرف طريقة عرض الـ Labels الخاصة بالـ Subjects و الـ Objects عندنا. إليك بعض الأدوات

1) الأمر ls

```
[root@KING-security4arabs ~]# ls -Z
-rw-------  root root system_u:object_r:user_home_t    anaconda-ks.cfg
-rw-r--r--  root root root:object_r:user_home_t        install.log
-rw-r--r--  root root root:object_r:user_home_t        install.log.syslog
```

2) الأمر id و secon

```
[root@KING-security4arabs ~]# id -Z
root:system_r:unconfined_t:SystemLow-SystemHigh
```

```
[root@KING-security4arabs ~]# secon
user: root
role: system_r
type: unconfined_t
sensitivity: SystemLow
clearance: SystemHigh
mls-range: SystemLow-SystemHigh
```

3) الأمر ps

```
[root@KING-security4arabs ~]# ps -axZ | grep httpd
Warning: bad syntax, perhaps a bogus '-'? See /usr/share/doc/procps-3.2.7/FAQ
system_u:system_r:httpd_t        2375 ?        Ss     0:00 /usr/sbin/httpd
system_u:system_r:httpd_t        2376 ?        S      0:00 /usr/sbin/httpd
system_u:system_r:httpd_t        2377 ?        S      0:00 /usr/sbin/httpd
system_u:system_r:httpd_t        2378 ?        S      0:00 /usr/sbin/httpd
system_u:system_r:httpd_t        2379 ?        S      0:00 /usr/sbin/httpd
system_u:system_r:httpd_t        2390 ?        S      0:00 /usr/sbin/httpd
system_u:system_r:httpd_t        2391 ?        S      0:00 /usr/sbin/httpd
system_u:system_r:httpd_t        2392 ?        S      0:00 /usr/sbin/httpd
system_u:system_r:httpd_t        2393 ?        S      0:00 /usr/sbin/httpd
root:system_r:unconfined_t:SystemLow-SystemHigh 6964 pts/0 R+   0:00 grep httpd
```

4) الأمر mkdir و الأمر install (يجب أن نكتب الـ Security context كاملة)

```
mkdir -Z user_u:object_r:user_home_dir_t Security4Arabs
```

5) الأمر find

```
[root@KING-security4arabs ~]# find /home/ -context '*:httpd_*_content_t'
/home/binary/public_html
/home/binary/public_html/index.html
/home/ba7ith/public_html
/home/ba7ith/public_html/index.html
/home/KING/public_html
/home/KING/public_html/index.html
```

الأرشفة و ضغط الملفات

من الطبيعي أن الملفات تورث الـ Label الخاص بمجلدها أو بمكانها و يتغير الـ label عند أرشفت الملفات و نقلها من مكان لمكان ذو Label مختلف في جزء type الملف. قد نحتاج عند أرشفة الملفات أن نقل معها نفس الـ Labels التي على الـ Objects و هذا يحدث كثيرا.

مثال لطريقة الأرشفة التقليدية.

```
[root@KING-security4arabs ~]# pwd
/root
[root@KING-security4arabs ~]# ls -Z anaconda-ks.cfg
-rw-------  root root system_u:object_r:user_home_t    anaconda-ks.cfg
[root@KING-security4arabs ~]# tar -czf anaconda-ks.cfg.tar.gz anaconda-ks.cfg
[root@KING-security4arabs ~]# ls -Z
-rw-------  root root system_u:object_r:user_home_t    anaconda-ks.cfg
-rw-r--r--  root root root:object_r:user_home_t        anaconda-ks.cfg.tar.gz
[root@KING-security4arabs ~]# ls -Z anaconda-ks.cfg.tar.gz
-rw-r--r--  root root root:object_r:user_home_t        anaconda-ks.cfg.tar.gz
[root@KING-security4arabs ~]# mv anaconda-ks.cfg.tar.gz /tmp/ ; cd /tmp
[root@KING-security4arabs tmp]# pwd
/tmp
[root@KING-security4arabs tmp]# ls -Z anaconda-ks.cfg.tar.gz
-rw-r--r--  root root root:object_r:user_home_t        anaconda-ks.cfg.tar.gz
[root@KING-security4arabs tmp]# tar -xzf anaconda-ks.cfg.tar.gz
[root@KING-security4arabs tmp]# ls -Z anaconda-ks.cfg
-rw-------  root root root:object_r:tmp_t              anaconda-ks.cfg
```

أرأيت!! كان الـ **type** الخاص بملف anaconda-ks.cfg في منزل الجذر هو user_home_t و بعد ضغطه و نقله إلى مجلد /tmp أصبح tmp_t.

هذه مشكلة عندما تحاول أرشفة ملفات كثيرة باختلاف الـ labels الخاص بها إلى مكان آخر. لكن بفضل الله يوجد طرق أفضل لهذا الغرض ,, تابع

سأعرض لكم الأدوات و طرية حفظ الـ label معها:

الأدوات:

1) tar و هي موجوده بشكل افتراضي و تدعم الـ Label من الإصدار RHEL v4 update 2 فما بعده

```
tar --selinux -cf anaconda-ks.cfg.tar.gz anaconda-ks.cfg
```

بعد نقل الملف المضغوط إلى **tmp** سنفك ضغطه

```
tar --selinux -xf anaconda-ks.cfg.tar.gz
```

الأن لنعرض الـ label للملف anaconda-ks.cfg لعد فك ضغطه في مجلد tmp

```
[root@KING-security4arabs tmp]# ls -Z anaconda-ks.cfg
-rw-------  root root system_u:object_r:user_home_t    anaconda-ks.cfg
```

2) star و هذه الأداة ليست موجودة بشكل افتراضي و يجب تركيبها

لضغط الملف:

```
star -xattr -H=exustar -c -f  anaconda-ks.cfg.star  anaconda-ks.cfg
```

فك الضغط

```
star -xattr -H=exustar -x -f  anaconda-ks.cfg.star
```

3) rsync و هي الأن تدعم الـ Label بشكل كامل و موجودة بشكل افتراضي

نعرف مميزات هذا البرنامج الرائع في النقل و النسخ نعم هو يقوم بنفس ما تقوم به الأوامر cp و scp لكنه يدعم استكمال النقل و لا يعيد النقل من البداية لهذا كانت الحاجة الماسة له في أشياء كثيرة و ها نحن نحتاج له هنا.

لأرشفة و نقل و الاحتفاظ بالـ label لملف

```
rsync  -avHPAX   anaconda-ks-cfg  10.0.0.80:/tmp
```

تطبيق عملي شامل

سيكون خادم الويب هو بطل تطبيقاتنا العملية لسهولته و لمعرفة الجميع به و باحتياجاته و طريقة عمله.

ستكون خطواتنا كالتالي:

سأفترض أن الـ SELinux يعمل في وضع permissive

إن لم يكن في وضع permissive اذهب إلى ملف /etc/selinux/config و عدلها ثم أعد التشغيل

1. تنصيب الأباتشي على CentOS 5.4
   1.1. عرض شكل الـ labels الأساسية للأباتشي في ملف file_contexts
   1.2. عرض الـ Label الخاص بخدمة الأباتشي
   1.3. عرض الـ Label الخاص بالمجلد الرئيسي للأباتشي.
   1.4. تشغيل صفحة html من مجلد الأباتشي الافتراضي.

2. إعداد الأبتاشي ليعمل على مجلد المستخدمين في مجلد اسمه public_html
   2.1. إعطاء التصاريح الصحيح لمحتويات المجلد publci_html وعرض صفحة html بناء على اسم المستخدم

3. إعطاء الـ Security Context الصحية للمحتويات public_html

4. إجبار الـ SELinux على السماح للأباتشي بالوصول إلى مجلدات المستخدمين.

لنبدأ ,,

1. تنصيب الأباتشي على CentOS 5.4

```
yum -y install httpd ; service httpd start ; chkconfig httpd on
```

1.1. عرض شكل الـ labels الأساسية للأباتشي في ملف file_contexts

grep httpd /etc/selinux/targeted/contexts/files/file_contexts | less

ستجد Context كثيرة و أهمها

File and Directory Type/Domain	Security Contexts
Excusable	system_u:object_r:httpd_exec_t
Configuration and System content	system_u:object_r:httpd_sys_content_t
Log Files	system_u:object_r:httpd_log_t
Scripts	system_u:object_r:httpd_script_exec_t
User Content	system_u:object_r:httpd_user_content_t

1.2. عرض الـ Label الخاص بخدمة الأباتشي

```
[root@KING-security4arabs ~]# ps -axZ | grep httpd
Warning: bad syntax, perhaps a bogus '-'? See /usr/share/doc/procps-3.2.7/FAQ
root:system_r:httpd_t           19581 ?        Rs     0:00 /usr/sbin/httpd
root:system_r:httpd_t           19583 ?        S      0:00 /usr/sbin/httpd
root:system_r:httpd_t           19584 ?        S      0:00 /usr/sbin/httpd
root:system_r:httpd_t           19585 ?        S      0:00 /usr/sbin/httpd
root:system_r:httpd_t           19586 ?        S      0:00 /usr/sbin/httpd
root:system_r:httpd_t           19587 ?        S      0:00 /usr/sbin/httpd
root:system_r:httpd_t           19588 ?        S      0:00 /usr/sbin/httpd
root:system_r:httpd_t           19589 ?        S      0:00 /usr/sbin/httpd
root:system_r:httpd_t           19590 ?        S      0:00 /usr/sbin/httpd
root:system_r:unconfined_t:SystemLow-SystemHigh 19592 pts/0 R+   0:00 grep httpd
```

كمان نرى فإن خدمة الأباتشي(httpd) مقيّدة في Domain إسمه: httpd_t

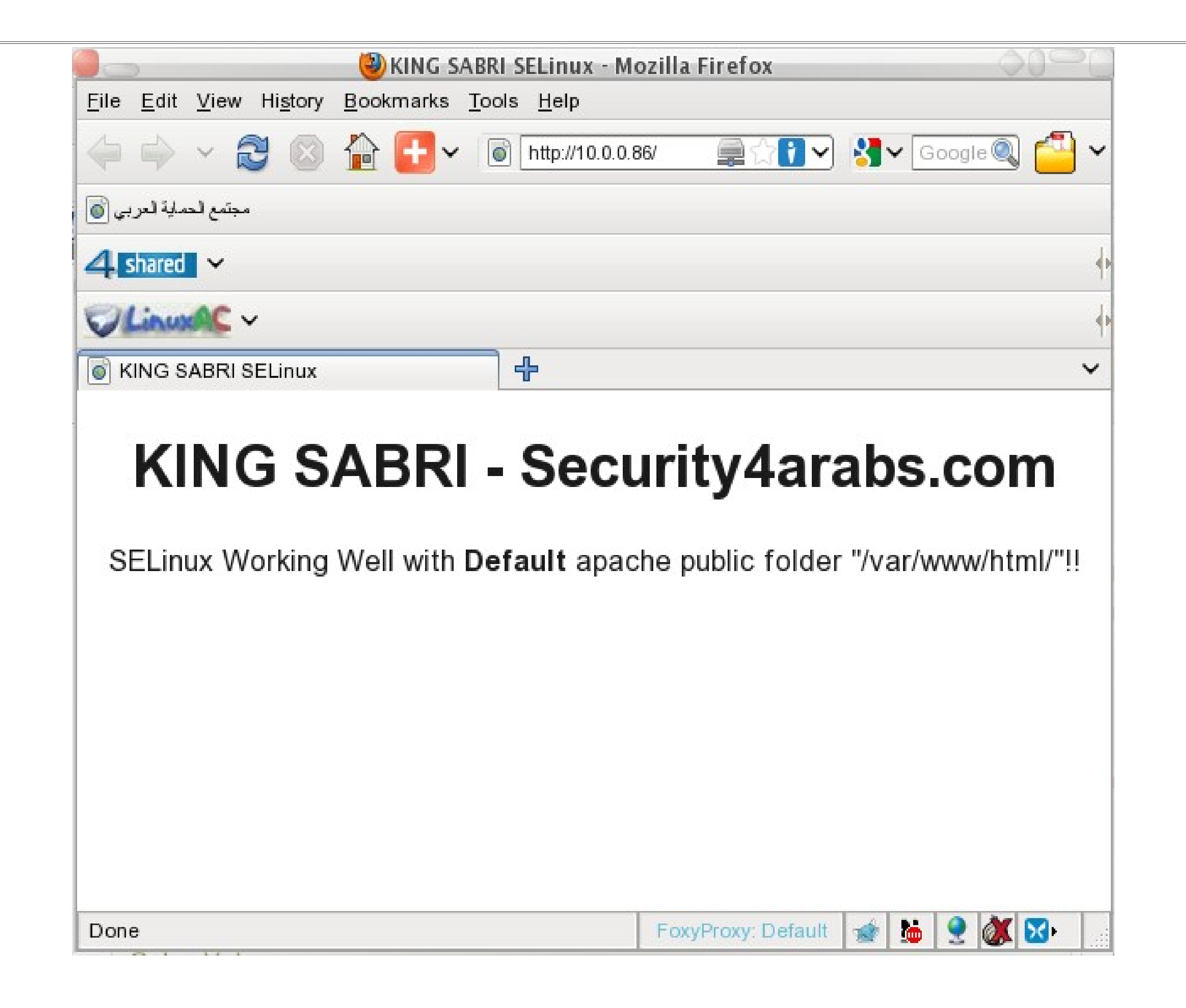
1.3. عرض الـ Label الخاص بالمجلد الرئيسي(الافتراضي) للأباتشي.

```
[root@KING-security4arabs ~]# ls -ldZ /var/www/html/
drwxr-xr-x  root root system_u:object_r:httpd_sys_content_t /var/www/html/
```

إذن هي تأخذ Label صحيح من البداية لأن هذا المكان مُعرّف في الـ Policy الموجودة من البداية

1.4. تشغيل صفحة html من مجلد الأباتشي الافتراضي.

أنشأنا ملف index.html في المجلد "var/www/html" , سنحاول تصفحها الأن


KING SABRI SELinux - Mozilla Firefox
File
Edit
View
History
Bookmarks
Tools
Help
http://10.0.0.86/
Google
مجتمع الحماية العربي
4 shared
LinuxAC
KING SABRI SELinux
KING SABRI - Security4arabs.com
SELinux Working Well with Default apache public folder "/var/www/html/"!!
Done
FoxyProxy: Default

2. إعداد الأبتاشي ليعمل على مجلد المستخدمين في مجلد اسمه public_html

نفتح ملف إعدادات الأباتشي

vim /etc/httpd/conf/httpd.conf

و نقوم بتهميش السطر "UserDir disable " و فك التهميش عن " #UserDir public_html "

```
    # UserDir disable

    UserDir public_html
```

و نعيد تشغيل الأباتشي

```
service httpd restart
```

2.1 إعطاء التصاريح الصحيح لمحتويات المجلد publci_html وعرض صفحة html بناء على اسم المستخدم

الأن سنقوم بإنشاء مجلد(إنشاء تقليدي) باسم public_html في مجلد المستخدم KING أيضا سنضع ملف index.html في المجلد الجديد

```
mkdir /home/KING/public_html
```

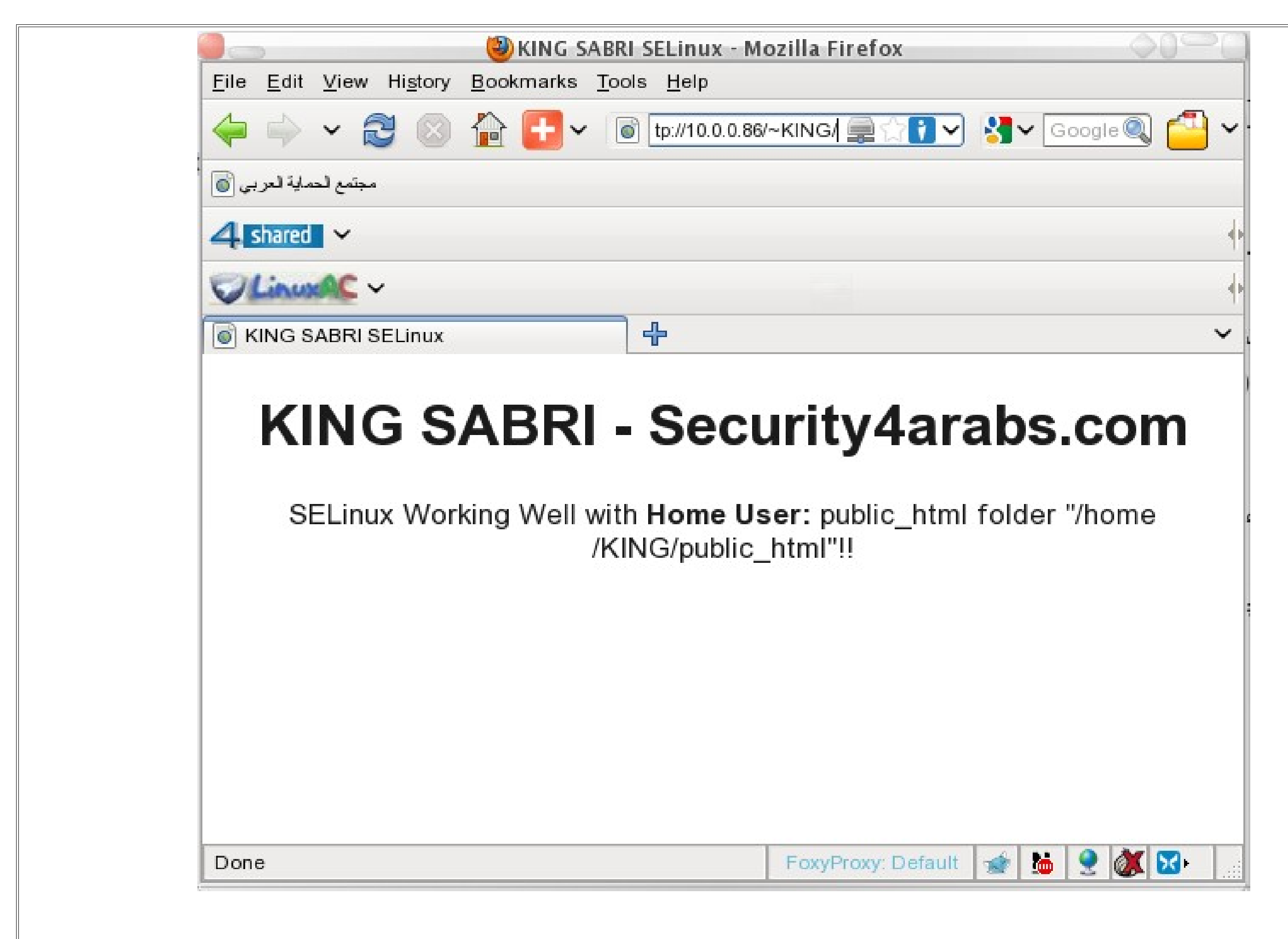
وسنقوم بإعطاء التصريح المناسب (DAC) لمجلد الـ home/KING ومحتوياته لأننا نبهنا أن DAC يتم تطبيقه قبل الـ MAC فإن قام DAC بالمنع فلا حاجة للـ MAC بأن ينظر في أمر المجلد أما لو قام DAC بالسماح وقتها ينظر MAC في أمر المجلد.

```
 chmod a+x /home/KING/
```

نبهت في بداية التطبيق أنني أفترض أن الـ SELinux يعمل في وضع permissive

الأن لنقم بتصفح الموقع بالمستخدم (مرة أخرى SELinux ليس في وضع enforcement)

هكذا (http://10.0.0.86/~KING/)



سنشغل الأن الـ SELinux في وضع enforcement (دون أي تعديل أخر) و سنرى ماذا سيحدث

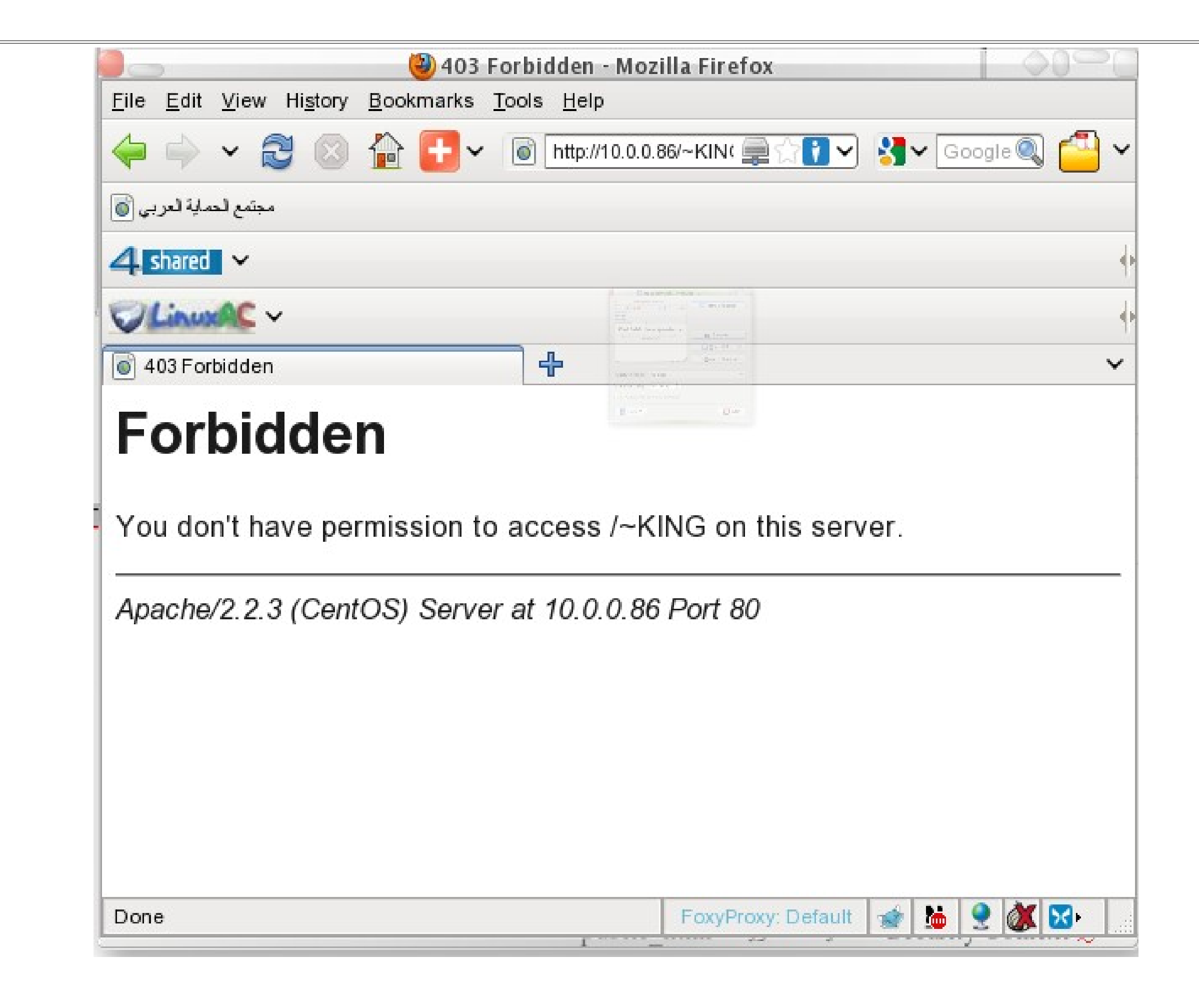
نفذ الأمر

```
echo 1 > /selinux/enforce
```

حيث هذا الأمر يجعل وضع SELinux هو enforcement لكن سيرجع إلى حالته الأصلية بعد أو إعادة تشغيل للنظام

اعمل تحديث للصفحة الأن

ستكون النتيجة:



لماذا ؟؟

لنتفقد الـ labels للـ Objects أولا

```
[root@KING-security4arabs ~]# ls -Z /home/KING/
drwxr-xr-x  root root root:object_r:user_home_t       public_html
```

إذن ما العمل ؟!

الحل في الخطوتين التاليتين:

3. إعطاء الـ Security Context الصحية للمحتويات public_html

4. إجبار الـ SELinux على السماح للأباتشي بالوصول إلى مجلدات المستخدمين.

هل لاحظت الـ label الخاص بمجلد public_html إن هذا الـ Object الأن تحت حماية type اسمه user_home_t و هو مختلف تماما عن الـ domain/type الخاص بالأباتشي المحددة في الـ Policy ,, لهذا السبب قامت الـ Role بمقارنة Domain الأباتشي مع الـ type الخاص بالمجلد فلم يتطابقا فمنعت وصول الأباتشي لهذا الـ Object

الحل هو تغير الـ label الخاص بمجلد الـ public_html و محتوياته بما يناسب الأباتشي (عرفنا ما يناسب الأباتشي في الخطوة **1.1**)

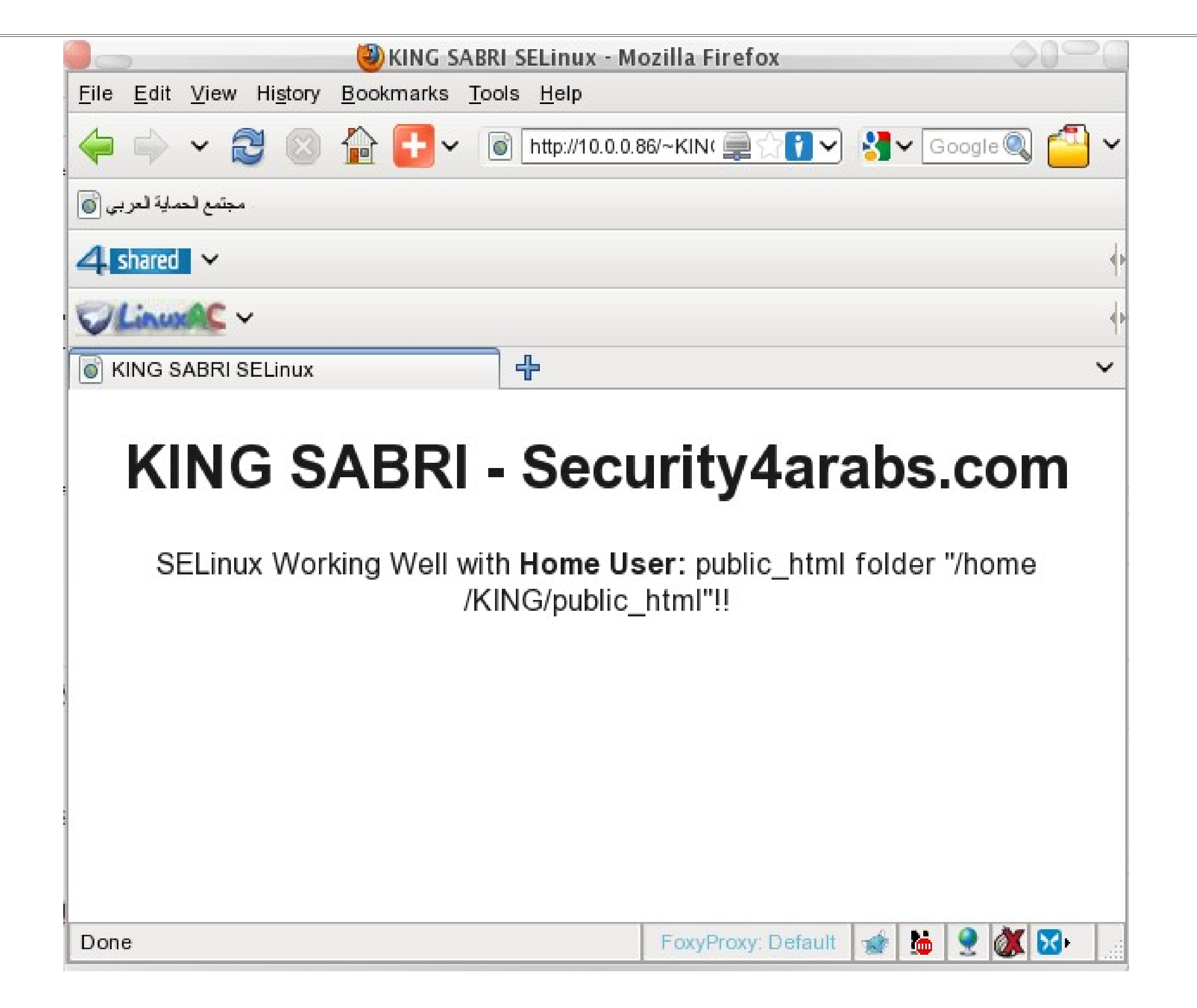
نفذ الأمر التالي**(**لإعطاء Security Context صحيحة**)**:

```
chcon -R -t httpd_user_content_t /home/KING/public_html
```

ثم نفذ الأمر التالي **(**لإجبار الـ SELinux على السماح للباشتي بالموصول إلى مجلدات المستخدمين**)**:

```
setsebool -P httpd_enable_homedirs on
```

الأن اعمل تحديث للصفحة مرة أخرى:



نعم,, إنها تعمل الآن !!

حتى هنا أكون قد أنهيت الباب الأول.

ملاحظة: إن لم تفهم المثال جيدا فليست مشكلة لأن هذا المثال سيتم شرح تفاصيله في الأبواب القادمة



الباب الثاني: استخدام الـ SELinux

أوضاع الـ SELinux

هناك شيئن سنتكلم عن أوضاعهما سواء بالتفصيل أو الإيجاز. و هما:

1. وضع عمل الـ SELinux
2. وضع عمل الـ Policy في SELinux

في البداية يجب أن نعرض ملف إعدادات الـ SELinux لنشرح عليه و مكانه في:

```
 /etc/selinux/config
```

و أيضا

```
/etc/sysconfig/selinux
```

يأتي الملف بهذا الشكل:

```
# This file controls the state of SELinux on the system.
# SELINUX= can take one of these three values:
#       enforcing - SELinux security policy is enforced.
#       permissive - SELinux prints warnings instead of enforcing.
#       disabled - SELinux is fully disabled.
SELINUX=enforcing
# SELINUXTYPE= type of policy in use. Possible values are:
#       targeted - Only targeted network daemons are protected.
#       strict - Full SELinux protection.
SELINUXTYPE=targeted
```

1. وضع عمل الـ SELinux

نرى في الملف ثلاثة أوضاع لحالة عمل الـ SELinux

a. حالة : enforcing

و هنا يعمل الـ SELinux بشكل كامل و يقوم بتطبيق الـ Policies و السماح و المنع و المراقبة و تسجيل لأحداث بشكل متكامل

b. حالة : permissive

و هنا يعمل الـ SELinux في وضع المراقبة فقط أي يقوم بتطبيق الـ Policies و تسجيل الأحداث ولكن دون المنع و السماح فهذا الوضع يستخدم للمراقبة و حل المشاكل التي تتعلق بـ SELinux و لاختبار البرامج الجديدة و مدى توافقها مع عمل الـ Policies الموجودة.

c. حالة: disabled

و هنا لا يعمل الـ SELinux على الإطلاق و يكون مغلق تماما

2. وضع عمل الـ Policy في SELinux

أيضا لتطبيق الـ Policiese هناك ثلاثة أوضاع

a. حالة : targeted

حيث هي الحالة المهمة لدينا في هذا الكتاب و هي التي نشرح عليه. في هذه الحالة يتم توجيه الـ Policy إلى الخدمات و أنواع الـ Subject و Object المعروفة, أما الغير معروفة فهي غير محمية

- تستطيع تحميل الـ Policy الخاصة بها

```
yum -y install selinux-policy-targeted
```

ستجد الملف التنفيذي له في/etc/selinux/targeted/policy/

- تستطيع تحميل أدوات تطوير الـ Policy

```
yum -y install selinux-policy-devel
```

ستجد الأدوات في /usr/share/selinux/devel/

b. حالة : strict

هذه الحالة تقيد جميع ما في النظام و تطبق عليه الـ Policy و هي معقدة جدا في إدارتها

- تستطيع تحميل الـ Policy الخاصة بها

```
yum -y install selinux-policy-strict
```

c. حالة : mls – Multi Level Security

كما هو واضع من اسمها فهي تستخدم في التطبيقات العسكرية غالبا , و تضيف على الـ Labels جزء الـ Sensitivities و الـ Categories و التي أشرنا إليها في الباب الأول

- تستطيع تحميل الـ Policy الخاصة بها

```
yum -y install selinux-policy-msl
```

التحكم بالـ SELinux

نعني بالتحكم هو كيفية فتح و إغلاق و تحويل وضع تشغيل الـ SELinux

هناك عدة طرق سأذكر

1. من ملف إعدادات الـ SELinux **(يحتاج إعادة تشغيل)**

```
 /etc/selinux/config
```

و أيضا

```
/etc/sysconfig/selinux
```

يو غير الوضع إلى ما تريد (enforcing ,permissive , disabled) من هذا السطر

```
SELINUX=enforcing
```

2. من ملف الإقلاع GRUB **(يحتاج إعادة تشغيل)**

تجده في :

```
/boot/grub/grub.conf
```

- لتشغيل أو إغلاق الـ SELinux أضف التالي في نفس سطر الكيرنيل

selinux=1 or 0

حيث: 0 = تعطيل . و 1= تشغيل

- لتشغل الـ SELinux في وضع enforcing أو permissive

enforcing=1 of 0

حيث: 0= permissive . و 1=enforcing

مثال :

```
kernel /boot/vmlinuz-2.6.18-164.15.1.el5 ro root=LABEL=/  selinux=1
```

3. الأمر setenforce **(لا يحتاج إعادة تشغيل)**

يفيد في التحول من enforcing إلى permissive و العكس (لا يقوم بتعطيل SELinux)

مثال:

```
setenforce 1
```

4. من ملف selinux/enforce/ **(لا يحتاج إعادة تشغيل)**

```
echo "1" > /selinux/enforce
```

5. من الواجهة الرسومية

```
system-config-selinux
```

سياق الملفات (File Context)

1) الأمر mv

عند نقل فإن جميع ما يتعلق بالملف/المجلد ينتقل معه لهذا لن يسبب لنا أي مشكلة

2) الأمر cp

يختلف الأمر cp عن mv لأن الأخر يقوم بنقل الملف فعليا أما cp فإنه يقوم بإنشاء ملف جديد في المكان الجديد و من ثم ينقل محتوياته و بما أننا أنشانا ملف جديد إذن سيأخذ labels جديه بناء على مكان

لتفادي هذا نستخدم

```
cp -a
```

أو نحدد label بأنفسنا كالتالي,,

```
[root@KING-security4arabs ~]# cp -Z system_u:object_r:file_t /etc/hosts hosts
```

إعادة سياق الملفات (Relabel Files)

قلنا أننا سنحتاج إلى تغير الـ Labels بما يناسب احتياجاتنا -كما احتجناها مع الأباتشي-(و بما يتوافق مع الحماية في نفس الوقت). لهذا كان لزامان علينا ذكر هذه النقطة السريعة.

هناك طريقتين تستطيع بهم عمل relabeling للملفات:

1) الأمر chcon

لتغير type/domain ملف

```
chcon -t typ_name_t  fileNme
```

لتغير type/domain مجلد و محتوياته

```
chcon  -R -t typ_name_t  fileNme
```

تستطيع أن تغير الـ user_id بتغير "-t" إلى "-u"

تستطيع أن تغير الـ role بتغير الـ "-t" إلى "-r"

لو كان عندك ملفين أحدهما الـ Label صحيح و الأخر غير صحيح تستطيع أن تجعل الأول مرجع للثاني دون الاضطرار إلى كتابه الـ label

```
chcon  --reference ConrrectFile  NotCorrectFile
```

2) الأمر restorecon

هذا الأمر يفيد فقط إذا كانت هناك ملفات/مجلدات ذات label مختلف و تريد أن تجعل الـ label خاصتهم ترج إلى الوضع الافتراضي للماكن الذي يشغلونه.. مثلا نحن عدلنا سابقا labels في مجلد الـ home للمستخدم KING و لكن لا نذكر الـ label الافتراضي و نريد أن نعيد كل شيء كما كان فهنا يأتي عمل هذا الأمر

```
restorecon -Rv  /home/KING/public_html
```

و هنا فقط سيظهر لك ما هو الـ label الذي يجب أن يكون موجود دون تغير كالأمر السابق

```
restorecon -nv
```

إعادة سياق نظام الملفات (Relabel a filesystem)

غالبتنا نجعل الـ SELinux في وضع Disabled ولكن بعد هذا الكتاب سيقوم من استوعبوا كلامي جيدا بتشغيله على الأقل في وضع permissive.
لكن سنحتاج في البداية أن يقوم الـ SELinux بالمرور على جميل ملفات/خدمات النظام ليقوم بإعطاءها الـ label المناسب , طبعا بحسب ما يوجد في الـ policy.

من هنا نعرف أن تشغيل SELinux لا يعني أن الـ Labels موجودة بشكل صحيح أو كامل و يجب عمل relabeling فقط لمرة واحدة.

هناك طريقتين لذلك

1) ملف autorelabel

قم بإنشاء ملف مخفي اسمه autorelabel في المجلد الجذر

```
touch /.autorelabel ; reboot
```

عند إعادة التشغيل سترى التالي:

```
*** Warning -- SELinux targeted policy relabel is required.
*** Relabeling could take a very long time, depending on file
*** system size and speed of hard drives.
/sbin/setfiles:  labeling files under /
**********************_
```

سيحتاج إلى وقت حتى ينتهي (بناء على حجم النظام و الملفات) و بعدها سيعيد التشغيل ثم ستقلع بشكل طبيعي

2) الأمر fixfiles **(لا يحتاج إعادة تشغيل)**

هذا الأمر قد يشبه في عمله chcon لكن chcon يشترط أن يكون هناك label موجودة أصلا أما هذا في fixfiles فإنه يضع label مناسب للـ policy و إن لم يوجد الـ label على Subject/Object.

```
[root@KING-security4arabs public_html]# fixfiles  relabel
/sbin/setfiles:  labeling files under /
********************************************************matchpathcon_filespec_eval:
hash table stats: 51421 elements, 35565/65536 buckets used, longest chain length 3
/sbin/setfiles:  labeling files under /mnt/disk1
matchpathcon_filespec_eval:  hash table stats: 4 elements, 4/65536 buckets used,
longest chain length 1
/sbin/setfiles:  Done.
```

أيضا يحتاج بعض الوقت ,,

فقط اذكر الله في هذا الوقت

لا إله إلا الله وحده لا شريك له له الملك و له الحمد و هو على كل شيء قدير

عمل mount مع الـ SELinux

نحتاج إلى SELinux أيضا مع الأقراص الصلبة ذات نظام ملفات لا يدعم الـ SELinux labeling مثل (fat32,ntfs) و أيضا مع مساحات التخزين التي لا نستطيع الوثوق بها مثل الـ Floppy و الـ CD-rom أيضا قد نحتاج لعمل mount لقرص من نوع ext2,ext3,ext4 لكنه لا يحتوي على labels بسبب أن جهازه الأصلي لم يفعل SELinux كما تحتاج أيتضا لعمل mount للملفات المشاركة من نوع nfs و smbfs مثلا.

لنرى بعض الأمثلة على ذلك و ستستوعب الفكرة

عمل mount لملفات مشاركة بـ nfs

```
mount -t nfs -o context=user_u:object_r:user_home_t 10.0.0.99:/shares/homes  /home
```

عمل mount لـ CD-rom

```
mount -o fscontext=system_u:object_r:removable_t  /dev/cdrom /media/cdrom
```

بالطبع في المراحل المتقدمة تستطيع أن تعمل rule خاصة و Context خاصة بأي أشياء دخيله تحتاج ربطها بجهازك



الباب الثالث: السياسة الموجهة(خاص بـ ريدهات)

الخدمات المحمية بالـ Targeted Policy

بناء على وجهة نظر ريدهات فقد أعطت اهتمام أكثر لبعض الخدمات في الـ Targeted Policy الخاصة بها. وهي مصنفة كالتالي

- **Web Services**, وهي : httpd , squid
- **Name and network Services** , وهي **:** bind/named, nscd, dhcpd
- **Authentication Services** , وهي **:** ypbind, winbindd
- **Database Services** , وهي **:** postmaster/postgreSQL, mysql
- **Administrative Services** , وهي **:** syslogd, ntpd, snmpd

ملاحظة: اهتمام ريدهات بالخدمات السابقة لا يعني أنها لا تطور الـ Policy للخدمات الأخرى

التحكم في حماية الخدمات

لعرض الخدمات المحمية أو لتعطيل تفعيلها توجه إلى المجلد

```
/selinux/boolians
```

ستجدها مكتوبه بالطريقة بهذه الطريقة

```
serviceName_disable_trans
```

و تكتوي على عمودين قيمتي العمودين تكون ثنائية إما 0 أو 1 لو كانت القيمة = 0 إذن فالخدمة محمية. مثال

```
cat  /selinux/booleans/httpd_disable_trans
0 0
```

القيمة الأولى(اليسرى) لحالة حماية الخدمة الأن

القيمة الثانية(اليمنى) هي حلة الـ pending أي حالة حماية الخدمة بعد إعادة التشغيل

```
/selinux/commit_pending_bools
```

لإغلاق حماية خدمة معينة.

```
echo "1" > /selinux/booleans/named_disable_trans
```

أيضا تستطيع الحفاظ على قيمة الـ commit_pending

```
echo "1 0" > /selinux/booleans/named_disable_trans
```

ثم أعد تشغيل الخدمة

```
service named restart
```

تستطيع أن تسهل الأمر على نفسك و تستخدم الأداة setsebool كما فعلنا في الفصل الأول

```
setsebool named_disable_trans 1
```

أو

```
setsebool named_disable_trans on
```

هذا التغير سيعود كما كان بعد أول إعادة تشغيل(لأن حالة الـ pending لم تتغير) و لجعل التغير دائم استخدم المفتاح P- مع الأمر كالتالي

```
setsebool -P named_disable_trans on
```

هناك أمر بديل يقوم **بقلب/تبديل** القيم الموجودة في الـ Policy بغض النظر ما هي القيمة و هو الأمر toggelsebool

```
toggelsebool named_disable_trans
```

أيضا تستطيع استخدام الواجهة الرسومية من

```
system-config-selinux
```

لعرض حالة الخدمة فقط

```
[root@KING-security4arabs ~]# getsebool  named_disable_trans
named_disable_trans --> off
```

طبعا تكتبها كما هي مكتوبة في selinux/booleans و تستطيع عرض حالة أكثر من خدمة في نفس السطر مع فصلهم بمسافة عادية

خدمة الأباتشي – Apache

واحدة من أهم و أخطر الخدمات على النظام لأنها أكثر الخدمات تعرضا للعالم الخارجي بل هي صنعت لتكون للعالم الخارجي و غالبا ما تكون هي مدخل الاختراق الشبكات الداخلية.

و بناء على خطورتها و أهميتها فقد تم الاهتمام بكل أجزاءها و كل ما يتعلق بها ووضع Security Context لها كما عرضنا ذلك في التطبيق العملي في الباب الأول و هنا سأضع الجدول لكن بإيضاح لتعرف أماكن الملفات و Security Context لها بالطبق سنستخدم أمر ls -Z في ذلك.

File and Directory Type/Domain	Security Contexts
/usr/sbin/httpd	system_u:object_r:httpd_exec_t
/etc/httpd/conf/*	system_u:object_r:httpd_config_t
/var/log/httpd/*	system_u:object_r:httpd_log_t
/var/www/cgi-bin/*	system_u:object_r:httpd_script_exec_t
/etc/init.d/httpd	system_u:object_r:initrc_exec_t
/var/www/html/*	system_u:object_r:httpd_sys_content_t

تذكير: راجع ملف

```
/etc/selinux/targeted/contexts/files/file_contexts
```

محتويات الأباتشي

قد عرفنا في المثال العملي في نهاية الباب الأول أن مجلد var/www/html له context خاصة و مختلفة و جربنا أن نشغل الموقع من مجلد المستخدم دون تعديل الـ label و لكن العملية فشلت و نجحت فقط عندما قمنا بتعديل label مجلد ال public_html الخاص بالمستخدم

بهذه الطريقة:

```
chcon -R -t httpd_user_content_t /home/KING/public_html
```

أيضا هناك طريقة أسهل و هي أن نجعل المجلد الأساسي للأباتشي هو المرجع للمجلد الجديد(بالنسبه لـ Security Context)

```
chcon -R --reference=/var/www/html /home/KING/public_html
```

أو أننا نعطي الـ context الصحيحة منذ إنشاء مجلد الـ public_html

```
mkfir --context system_u:object_r:httpd_sys_content_t /home/KING/public_html
```

أظنك بدأت تشعر بمرونة الوضع ,,

إعدادات منطقية خاصة للأباتشي

من المعروف أن الأباتشي يصل إلى عدة أماكن و لتلبية طلبات المستخدم في عرض الصفحات. لكن هناك أماكن خطيرة مثل CGI فعن طريقها تستطيع تنفيذ أوامر مباشرة إلى السيرفر و عرفنا أن الأساس في SELinux هو المنع إذن CGI ممنوع أيضا بشكل افتراضي لمثل هذه الأشياء.

توجد قيم منطقية Boolean في SELinux كما ذكرنا من قبل و هي تحتوي على مفاتيح سريعة للتحكم بالـ Policy

مكان الملف الذي يكتوي القيم المنطقية(ذكرناه سابقا أيضا)

```
/selinux/booleans/
```

سنعرض قيم مهمة لك و يوجد الكثير يجب أن تقرأ عنها لتعرف ما يناسبك لتقوم بتعطيل المنع منهم

httpd_enable_cgi

قلنا أن الـ CGI خطير لأنه يسمح بتنفيذ الـ Scripts و الأوامر على السيرفر مباشرة

httpd_ssi_exec

Server Side Include هي نفس فكرة عمل الـ CGI – أوامر تستطيع تنفيذها على السيرفر من صفحة ويب.

httpd_enabled_homedirs

كما عرفناها سابقا و هي لنسمح للأبتاتشي أن يصل إلى مجلد المستخدم .

httpd_tty_comm

تسمح لخادم الويب أن يتواصل مع النظام عن طريق الأوامر console

تستطيع أن ترى جميع القيمة المنطقية للأبتاشي بهذا الأمر

```
getsebool -a | grep httpd
```

خدمة أسماء النطاقات DNS

كما الحال في خادم الويب, فإن خوادمه خدمة النطاقات DNS server مهمة جدا في حياتنا و تعاملاتنا الإلكترونية و أصبحت جزء لا يتجزأ منها كما أنها قد تحتوي على ثغرات و تتعرض لهجمات كأي خدمة أخرى.

من أحد أشكال عمل الـ DNS هو Dynamic DNS أو DDNS و سنفترض سناريو لوضع معين . عندنا خادم DNS رئيسي/Master و آخر فرعي/Slave حيث الرئيسي يتحكم بالنطاقات/Domains و الفرعي يقوم بنقل الـ Zones و هناك حلقة وصل دائمة بين الرئيسي و الفرعي حيث الفرعي يزود الرئيسي بكل الـ Zones الجديدة أولا بأول و هي متغيرة لهذا أصبح DDNS . إذن هناك احتمالية دائمة أن هناك من يستطيع تغير معلومات الـ DNS الرئيسي, فماذا لو كان من يغير معلومات الـ master أحد غير الـ slave الحقيقي؟

جمل الحماية و الجمل المنطقية للـ DNS

سأعرض لكم جمل الحماية المتعلقة بخدمة الـ DNS – BIND

File and Directory Type/Domain	Security Contexts
/usr/sbin/named	system_u:object_r:named_exec_t
/etc/named.conf	system_u:object_r:named_config_t
/var/named/*.zone	system_u:object_r:named_zone_t
/var/log/named	system_u:object_r:named_log_t
/etc/init.d/named	system_u:object_r:initrc_exec_t

و مثل الأبتاشي فهو يحتوي على قيم منطقية

named_disable_trans

لإغلاق الحماية من على الـ bind

named_write_master_zone

لكي تسمح بتحديث ملفات الـ zone في الخادم الرئيسي, حيث السماح بهذا يسمح لنا بعمل DDNS.

خدمات أخرى

dhcpd

mysqld

nscd

ntpd

squid

syslogd

winbind

ypbind

و هناك خدمات كثيرة تستطيع أن تجدها في:

RedHat

http://www.redhat.com/docs/manuals/enterprise/RHEL-4-Manual/selinux-guide/rhlcommon-chapter-0051.html

http://www.redhat.com/docs/en-US/Red_Hat_Enterprise_Linux/5.2/html/Deployment_Guide/sec-selinux-policy-targeted-rolesandusers.html

CentOS

http://www.centos.org/docs/5/html/Deployment_Guide-en-US/sec-sel-policy-targeted-oview.html

جمل الحماية لخدمات أخرى

سأعرض جميل الحماية لبعض الخدمات محالوة أن تألف أشكالها لزيادة مرونه تعاملك معها لاحقا

ملف إعدادات الخدمات التالية

File and Directory Type/Domain	Security Contexts
/usr/sbin/dhcpd	system_u:object_r:dhcpd_exec_t
/etc/dhcpd	system_u:object_r:dhcpd_etc_t
/sbin/portmap	system_u:object_r:portmap_exec_t
/usr/sbin/squid	system_u:object_r:squid_exec_t
/etc/squid/squid.conf	system_u:object_r:squid_conf_t
/usr/sbin/ntpd	system_u:object_r:ntpd_exec_t
/usr/sbin/snpd	system_u:object_r:snpd_exec_t
/etc/snmp/snmp.conf	system_u:object_r:snmp_etc_t
/sbin/syslogd	system_u:object_r:syslogd_exec_t



الباب الرابع: التتبع و حل المشاكل

تحديد المشكلة

علمنا أن الأصل في الـ SELinux هو المنع حيث سنواجه(في بداية الأمر) مشاكل متوقعة و غير متوقعة كثيرة, لكني أعتقد إن أي شخص يعتاد على وجوده فلن يستطيع الاستغناء عنه.

~ لو افترضنا أن النظام لم يقلع بشكل صحيح و كنت متأكد أن السبب هو SELinux , فادخل إلى single mode أو إلى rescue mode و ادخل إلى ملف الإقلاع grub.conf وضع القيمة enforcing=0 أو selinux=0 كما ذكرنا في التحكم في SELinux.

~ لو واجهت مشكلة في تسجيل الدخول بأحد المستخدمين, فتأكد من أن الـ labels لمجلدات المستخدمين صحيحة أو قد تكون ليس هناك أي labels و SELinux يرفض هذا. للتذكير يجب أن يكون الـ label لمجلد المستخدم (user_home_dir_t).

~ لو تعطل مع تطبيق أو برنامج معين و كان يعمل قبل تفعيل SELinux, قم بإعطاءه الـ label المناسب لطبيعة عمله.

~ لو تعطلت خدمة كانت تعمل قبل تفعيل SELinux , قم بعمل بإعادة تشغيل الخدمة بالطريقة التالية , (سيطلب منك كلمة مرور الجذر)

```
/usr/sbin/run_init  /etc/init.d/Service_name restart
```

حسنا ,, كيف تفكر عند وجود مشكلة بسبب SELinux بشكل عام ؟

1. اعرض رسالة الخطأ التي ظهرت في رسالة الـ avc

- اعرف من الرسالة ماهي الخدمة الموقوفة تجدها في جزء scontext

- ما هو توجه هذه الخدمة أو ماذا تريد أن تفعل؟ تجدها في جزء path و tclass فهذا الجزء يخبر في أي Object تريد أن تعدل هذا الـ subject و ستجد أيضا جزء ino و تعني iNode وهو حيث SELinux يصدر تقاريره بالاعتماد على الـ node الموجود في device.

تستطيع عرض الـ inode للملفات بـ (`ls -li`) أو البحث عنها بـ (`find  / -inum xxx -print`).

2. يجب تحديد ما هي الخدمة و ما عملها و لماذا تم منعها من SELinux و ستعرف ذلك أيضا من رسالة avc في جزء tcontext و عادة يكون فيها مشاكل الخدمات أو البرامج أو الملفات التنفيذية و الاسكربتات

3. إذا كنت تريد أن تحلل الـ Policy التي تعمل بها استخدم الأداة apol.

4. إذا رأيت أي عيوب في الـ policy أو كان لك اقتراحات عليها فالرجاء الإبلاغ عنها هنا (https://bugzilla.redhat.com/)

التعامل مع السجلات SELinux Auditing

يتم تخزين الأخطاء الصادرة من SELinux عن طريقة خدمة موجودة بشك افتراضي و اسمها **auditd** و تخزن الأخطاء في ملف

```
/var/log/audit/audit.log
```

فإن لم توجد الخدمة-لسبب أو لآخر- فإن الأحداث تسجل في

```
/var/log/messages
```

التعامل مع AVC

قبل التعامل معها , يجب أن نعرفها ,,

فما هي رسائل AVC ؟

هي اختصار لكمة (Advanced Vector Cache) و هي رسائل لا غنى عنها, في تقوم بإعطاءنا تقرير كامل و مفيد عن الخطأ, من أين و ما سببه و أين مكانه والرائع أيضا أنها تساعدك في حل المشكلة وليس فقط توضيع المشكلة حتى أنها تصل أنها تعطيك الأمر كامل لحل المشكلة.

سأعطيك مثال عن خطأ و سنقوم بتشريح و تفصيل الرسالة. الخطأ سببه محاولة الوصول إلى صفحة web .

```
type=AVC msg=audit(1273808351.267:175): avc:  denied  { getattr } for  pid=10586
comm="httpd" path="/home/KING/public_html" dev=hda1 ino=959060
scontext=root:system_r:httpd_t:s0 tcontext=root:object_r:user_home_t:s0 tclass=dir
```

رائع!!

لنشرح الرسالة:

```
type=AVC
```

هذا الجزء يخبرك أي نوع من أنواع الأخطاء هذا الخطأ لتسهيل البحث و قراءة الأخطاء .و عرفنا أنه من رسائل avc.

```
msg=audit(1273808351.267:175)
```

هذا الجزء يحتوي على ثلاثة معلومات:

الأولى: نوع الرسالة

```
msg=audit
```

حيث الرسالة تخرج من الـ Kernel على مستويات و أيضا هناك مؤشر يوجه الرسائل الخارجة من الـ kernel

الثانية: الوقت و التاريخ

```
1273808351.267
```

حيث يقوم بتسجيل التاريخ و الوقت بأجزاء من الثانية(Milliseconds) و تسمى هذه الصيغة بـ Epoch time

طبعا من الصعب عليك قراءة الوقت بهذا الشكل و لهذا,, إليك طريقة تحويل هذا الوقت إلى شكل مقروء

يكون الأمر كالتالي

```
date -d @EPOCH_TIME
```

في مثالنا,,

```
[root@KING-security4arabs ~]# date -d @1273808351.267
Fri May 14 06:39:11 AST 2010
```

و بالصدفة ,, هذا هو تاريخ ميلادي :D

الثالثة: الرقم التسلسلي للخطأ

```
175
```

و هو لتسهيل الإبلاغ و اكتشاف نوع الخطأ دون الحاجة إلى قراءة جميع الرسالة (تأتي بالخبرة)

```
comm="httpd"
```

هذا الجزء يخبرك على الأمر/الخدمة/الملف التنفيذي/المستخدم/ Subject الذي تم منعه و هنا نرى أنه أمر خدمة الأباتشي httpd.

```
 path="/home/KING/public_html" dev=hda1 ino=959060
```

هنا وضح مكان الـ Object الذي تمت حمايته مع مكانه على القرص الصلب و inode

```
scontext=root:system_r:httpd_t:s0
```

و هذا الجزء يبين لنا الـ Label أو Security Context للـ Subject الذي تم منعه

```
tcontext=root:object_r:user_home_t:s0
```

و بالمثل فقد وضّح لك هنا الـ Label الخاص بالـ Object الذي تمت حمايته

```
tclass=dir
```

أخيرا أحبً أن يوضح لك نوع الـ Object و هنا هو مجلد و ليس ملف و مما سبق عرفنا مساره "/home/KING/public_html"

حل المشاكل Troubleshooting

معرفة المشكلة و سببها يعتبر 70% من حل المشكلة, و لهذا قبل قبل أن نبدأ, يجب عليك تثبيت الحزمة التالية "**setroubleshoot**":

```
yum -y install setroubleshoot* ; service setroubleshoot start ; chkconfig
setroubleshoot on
```

هناك خدمة اسمها setroubleshootd و تقوم بمراقبة رسائل الـ avc ثم تقوم بإرسال تلك الرسائل إلى السجلات مع وضع حل مقترح أيضا وهو برنامج له واجهة رسومية أيضا

لاستخدامه من سطر الأوامر

```
sealert -a /var/log/audit/audit.log
```

ستكون المخرجات كالتالي:

```
--------------------------------------------------------------------------------
Summary:

SELinux is preventing the httpd from using potentially mislabeled files
(/home/KING/public_html).

Detailed Description:

SELinux has denied httpd access to potentially mislabeled file(s)
(/home/KING/public_html). This means that SELinux will not allow httpd to use
these files. It is common for users to edit files in their home directory or tmp
directories and then move (mv) them to system directories. The problem is that
the files end up with the wrong file context which confined applications are not
allowed to access.

Allowing Access:

If you want httpd to access this files, you need to relabel them using
restorecon -v '/home/KING/public_html'. You might want to relabel the entire
directory using restorecon -R -v '/home/KING/public_html'.

Additional Information:

Source Context                root:system_r:httpd_t
Target Context                root:object_r:user_home_t
Target Objects                /home/KING/public_html [ dir ]
Source                        httpd
Source Path                   /usr/sbin/httpd
Port                          <Unknown>
Host                          <Unknown>
Source RPM Packages           httpd-2.2.3-31.el5.centos.4
Target RPM Packages
Policy RPM                    selinux-policy-2.4.6-255.el5_4.4
Selinux Enabled               True
Policy Type                   targeted
```

```
MLS Enabled                   True
Enforcing Mode                Enforcing
Plugin Name                   home_tmp_bad_labels
Host Name                     KING-security4arabs
Platform                      Linux KING-security4arabs 2.6.18-164.6.1.el5 #1
                              SMP Tue Nov 3 16:18:27 EST 2009 i686 i686
Alert Count                   2
First Seen                    Fri May 14 06:39:11 2010
Last Seen                     Fri May 14 06:39:11 2010
Local ID                      cc319b72-4736-413f-bfb6-3ad4f71c55db
Line Numbers                  289, 290, 291, 292

Raw Audit Messages

type=AVC msg=audit(1273808351.267:175): avc:  denied  { getattr } for  pid=10586
comm="httpd" path="/home/KING/public_html" dev=hda1 ino=959060
scontext=root:system_r:httpd_t:s0 tcontext=root:object_r:user_home_t:s0 tclass=dir

type=SYSCALL msg=audit(1273808351.267:175): arch=40000003 syscall=195 success=no
exit=-13 a0=8126958 a1=bfe87d0c a2=575ff4 a3=8170 items=0 ppid=10584 pid=10586 auid=0
uid=48 gid=48 euid=48 suid=48 fsuid=48 egid=48 sgid=48 fsgid=48 tty=(none) ses=13
comm="httpd" exe="/usr/sbin/httpd" subj=root:system_r:httpd_t:s0 key=(null)
```

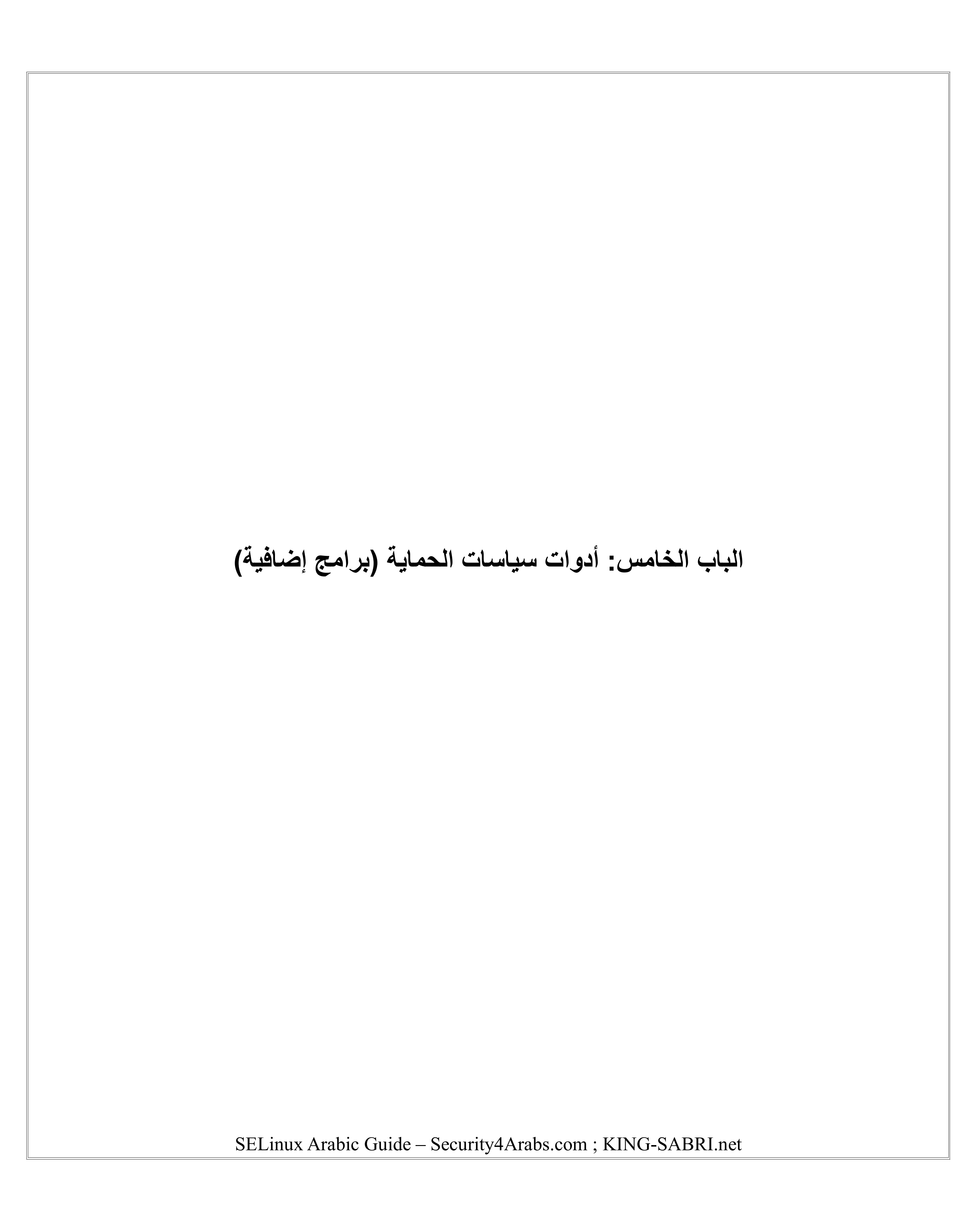
لو لاحظت ما كبرت خطه قليلا ....

ستجده الحل كاملا!!!

وقد قمت بتطبيقه أيضا و فعلا حل المشكلة.

```
[root@KING-security4arabs ~]# restorecon -R -v /home/KING/public_html/
restorecon reset /home/KING/public_html context root:object_r:user_home_t:s0-
>user_u:object_r:httpd_sys_content_t:s0
restorecon reset /home/KING/public_html/index.html context
root:object_r:user_home_t:s0->user_u:object_r:httpd_sys_content_t:s0
```

تنبيه: ليس شرطا أن يكون الحل معطى لك هو الآمن لك, فلك عقل تفكر به و أنت تعرف احتياجاتك جيدا.

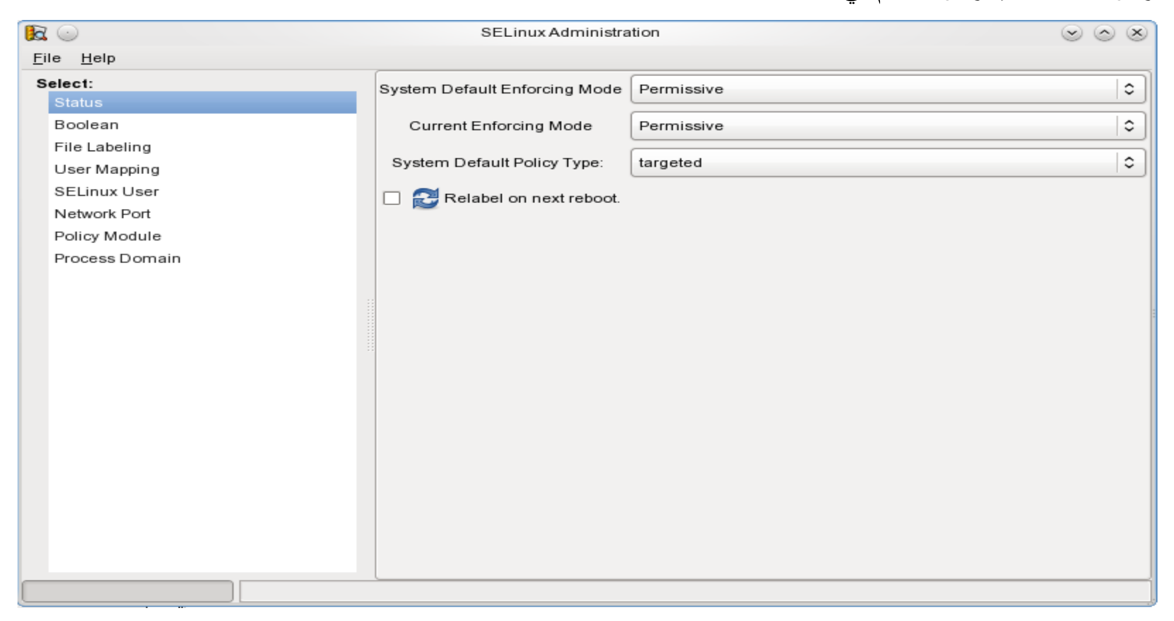


الباب الخامس: أدوات سياسات الحماية (برامج إضافية)

سأعرض عليكم بعض الأدوات المستخدمة في التعامل مع SELinux و وظيفة كل واحد دون إسهاب و أترك لك استكشافها و التعمق فيها.

الأداة system-config-selinux

و هي أداة بواجهة رسومية مهمتها إدارة و التحكم في الـ SELinux



الأداة seaudit

أداة من ضمن مجموعة أدوات **setools** لقراءة سجلات الـ SELinux وترتيب المخرجات حسب الرغبة. تعمل عبر سطر الأوامر و عبر الواجهة الرسومية

لتنزيل الحزمة

```
yum -y install setools*
```

استخدامها من سطر الأوامر

```
 seaudit-report /var/log/audit/audit.log
```

أو إخراج الأحداث في صفحة ويب

```
seaudit-report  --html -o selinuxLog.html /var/log/audit/audit.log
```

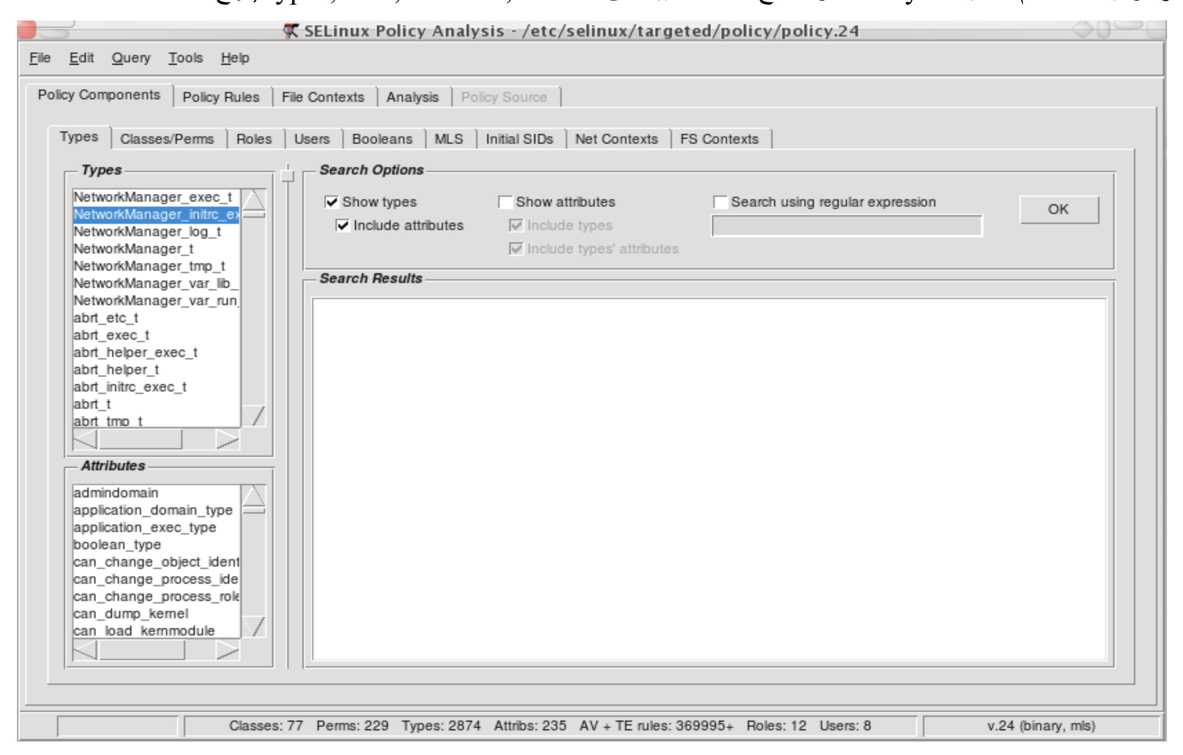
استخدامها من الواجهة الرسومية

```
seaudit -l /val/log/logFilePath.log
```

إن كان ملف الأحداث هو الافتراضي فلا داعي لكتابة المسار فقط اكتفي بتشغيله seaudit

الأداة apol

أداة ذات واجهة رسومية تستخدم لتحليل الـ Policy و تسمح بالبحث فيها عن الـ types,roles,booleans,SIDs , إلخ..



الأداة checkpolicy

أداة من سطر الأوامر تقوم بأتكد من صحة كتابة الـ Policy و خلوها من الأخطاء المنطقية و الكتابية.

الأداة sesearch

أداة من سطر الأوامر للبحث عن ملف بمعلومات الـ labels

```
sesearch  -a -t httpd_user_content /etc/selinux/targeted/policy/policy.21
```

الأداة sestatus

أداة من سطر الأوامر تقوم بعرض حالة عمل الـ SELinux كـ (disabled,enforcing ,permissive) و حالة , إصدار الـ Policy . تستطيع إعداد طريقة عرض الأمر و اختيار المعلومات التي تريد عرضها. من ملف /etc/sestatus.conf

```
sestatus -v
```

الأداة audit2allow

يعرض رسائل avc من نوع allow من ملف السجل

```
audit2allo -l /var/log/audit/audit.log
```

الأداة audit2why

أداة من سطر الأوامر تعرض لك لماذا قامت avc بإصدار الخطاء في السجل أي سبب الخطأ

```
audit2why <  /var/log/audit/audit.log
```

الأداة sealert

أمر يتصل بخدمه اسمها setroubleshootd و تقوم بمراقبة رسائل الـ avc ثم تقوم بإرسال تلك الرسائل إلى السجلات مع وضع حل مقترح أيضا وهو برنامج له واجهة رسومية أيضا

لاستخدامه من سطر الأوامر

```
sealert -a /var/log/audit/audit.log
```

الأداة avcstat

أداة من سطر الأوامر تعرض لك كم مرة قام SELinux باتخاذ إجراء و كم مرة تم أرسال avc هذه المعلومات يتم أخذها من ملف /selinux/avc/cache_stats

الأداة seinfo

أداة من سطر الأوامر تعرض لك إحصائيات عن كل صغيرة و كبيرة في policy و عدد ال types و المستخدمين و القيم المنطقية , إلخ,,

الأداة semanage

أداة هامة جدا من سطر الأوامر للتحكم بالـ policy و التعديل في الـ context على الـ Subjects/Objects على حد سواء

```
semanage login -l
semanage user -l
```



الخاتمة

في النهاية أسأل الله الكريم رب العرش العظيم أن يتقبل هذا الكتاب قبولا حسنا و أن ينفع به عباده و يزقنا أجره.
كما أنني أنوه أنه سيكون لهذا الكتاب جزء ثاني متقدم -إن شاء الله- أو قد أضيفه لاحقا على نفس الكتاب فيكون كتابا واحدا.

تحياتي و احترامي
صبري صالح
Security4arabs.com
KING-SABRI.NET